

Amendments to the Specification:

Please replace the paragraph beginning at page 3, line 26 with the following amended paragraph:

As shown in FIG. 2, a system 20 is divided into a primary system 36 and a recompilation system 38. Users manage the primary system 36. Recompilation system 38 monitors all relations in the primary system 36 and builds a representation parallel to the representation in the primary system 36. This representation includes interface levels 32 and dependency lists 34. Primary system 36 includes an interface 30. The interface 30 includes information in the global unit 12 that is referenced by a client 14 or indirect client 16. In this example, clients 14a, 14b, and 14c all depend on information included in interface 30. This information can include global components, the definition of attributes, and signatures of methods. A designer of the recompilation system 38 generates interface levels 32 (included in the recompilation system 38) during design time. Each interface level 32 corresponds to a subset of the global unit 12 and includes a dependency list 34. Each dependency list 34 includes associations to a list of clients 14 and indirect clients 16. The recompilation system 38 automatically determines the appropriate list for each client such that a representation of the client interface included in primary system 36 is generated in the recompilation system 38. For example, as shown in FIG. 2, clients 14a and 14b are associated with dependency list 34a while client 14c is associated with dependency list 34b. The system 10 uses the dependency list 34 to determine clients 14 and indirect clients 16 to mark for recompilation if a change is made to a global component on which the indirect client 16 depends.

Please replace the paragraph beginning at page 5, line 28 with the following paragraphs:  
[a paragraph break has been added]

In another example, all interface levels 32 include references to particular information in the system 10. For example, all clients 14 and indirect clients 16 can depend on an existence of a common variable X. Thus, if a user renames variable X to be variable Y, all clients 14 and

indirect clients 16 need to be recompiled. Thus, if a change affects all interface levels 32, all clients 14 and indirect clients 16 can be marked for recompilation simultaneously.

In another example, the global unit 12 includes three interface levels. A first level includes clients 14 and indirect clients 16 with a strong dependency on the interface. This level incorporates layout information of the interface 30 and the system 10 marks a client 14 or indirect client 16 for recompilation after every change in the interface. A second interface level includes clients 14 and indirect clients 16 with a strong dependency on an interface component (e.g. a reference to name of a component interface and one of the interface components). This interface level is associated with layout information for the component and the system 10 marks a client 14 or indirect client 16 for recompilation only if the component is changed. For example, if the name of the component changes, the component is deleted, or the layout of the component changes. A third interface level includes clients with a reference to the interface itself, but not to a component (i.e. a weak reference). The client depends only on the existence of the interface and the system 10 marks the client for recompilation only if the interface itself is changed.